



PATENT ABSTRACTS OF JAPAN

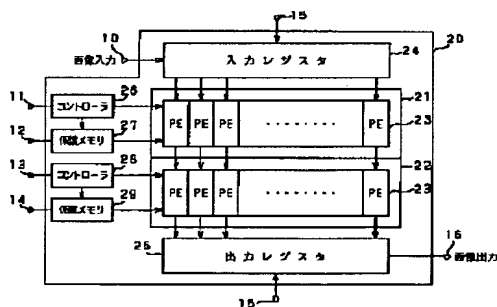
(11) Publication number: **10134176 A**(43) Date of publication of application: **22.05.98**

(51) Int. Cl.

G06T 3/40**G06T 5/20****H04N 1/387****H04N 7/01**(21) Application number: **08287103**(22) Date of filing: **29.10.96**(71) Applicant: **SONY CORP**(72) Inventor:
NAKAMURA KENICHIRO
KUROKAWA MASUYOSHI
KANO MAMORU
AOYAMA KOJI**(54) METHOD AND DEVICE FOR IMAGE SIGNAL PROCESSING****(57) Abstract:**

PROBLEM TO BE SOLVED: To easily vary the number of taps and a conversion ratio in a short time by automatically setting a filter program and a filter coefficient according to filter specification information and performing filter operation.

SOLUTION: Pixel data of one horizontal line after being inputted to an input register 24 in order from an image input terminal 10 are supplied to an upper processor element(PE) group 21 in parallel. The PE group 21 performs filter operation processing for respective pixel data that respective element processors 23 have received by using a control instruction from a controller 26 and coefficient data from a coefficient memory 27. The processing result of the upper PE group 21 is inputted to the lower PE group 22 as it is in parallel and filter operation processing is further performed when necessary. The processing result of the PE group 22 is inputted to an output register 25 in parallel and outputted, pixel by pixel, in order. The processing like this is repeated as many times as vertical lines to perform two-dimensional image processing.



THIS PAGE BLANK (USPTO)

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平10-134176

(43)公開日 平成10年(1998) 5月22日

(51)Int.Cl.⁸
G 0 6 T 3/40
5/20
H 0 4 N 1/387
7/01

識別記号

1 0 1

F I

G 0 6 F 15/66 3 5 5 C
H 0 4 N 1/387 1 0 1
7/01 C
G 0 6 F 15/68 4 0 0 J

審査請求 未請求 請求項の数7 O L (全 21 頁)

(21)出願番号 特願平8-287103

(22)出願日 平成8年(1996)10月29日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 中村 憲一郎

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(72)発明者 黒川 益義

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(72)発明者 加納 健

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(74)代理人 弁理士 小池 晃 (外2名)

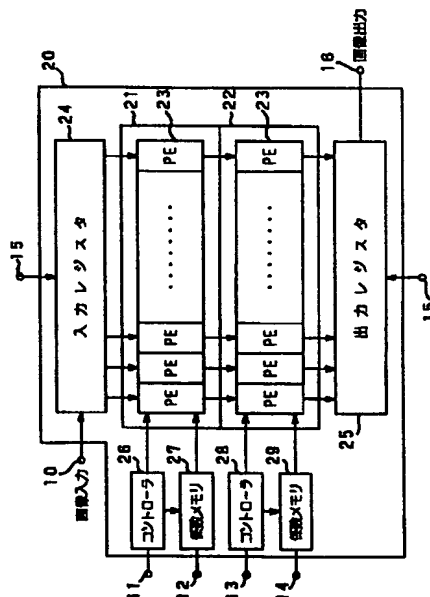
最終頁に続く

(54)【発明の名称】 画像信号処理方法及び装置

(57)【要約】

【課題】 タップ数や変換比率を変更したい時に、容易かつ短時間でそれらの変更を可能にし、また、複数の変換比率にて画素数変換を行う必要がある場合でもそれぞれの変換比率に対応する複数用意する必要を無くする。

【解決手段】 2次元画像に対してフィルタ演算を施す画素数変換装置であり、フィルタ仕様情報に基づいて発生された水平、垂直両方向用のフィルタプログラム及びフィルタ係数が供給されるコントローラ26、28及び係数メモリ27、29と、水平ラインの画素データをシリアル/パラレル変換する入力レジスタ24と、入力レジスタ24の出力画素データに水平フィルタ演算を施すプロセッサエレメント群21と、プロセッサエレメント群21の出力画素データに対して垂直フィルタ演算を施すプロセッサエレメント群22と、プロセッサエレメント群22の出力データをパラレル/シリアル変換する出力レジスタ25とを有する。



【特許請求の範囲】

【請求項1】 第1の方向及び当該第1の方向と交差する第2の方向に配列する複数画素からなる2次元画像に対して、フィルタ演算を施す画像信号処理方法において、

フィルタ仕様情報に基づいて、上記第1の方向と第2の方向の両方向用のフィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生し、

上記2次元画像を構成する上記第1の方向の画素順で、順次シリアルに入力した画素データを当該第1の方向の画素列毎にパラレルに変換し、

上記パラレルに変換された上記第1の方向の画素列の各画素データに対して、当該第1の方向用のフィルタプログラム及びフィルタ係数を用いた所定のフィルタ演算処理を施し、

当該所定の演算処理がなされた上記第1の方向の画素列の各画素データに対して、必要に応じて更に上記第2の方向用のフィルタプログラム及びフィルタ係数を用いた所定のフィルタ演算処理を施し、

得られたパラレル画素データを上記第2の方向の画素順に順次シリアル出力することを特徴とする画像信号処理方法。

【請求項2】 フィルタ仕様情報に基づいて上記2次元画像の拡大または縮小の設定を行い、

上記2次元画像の拡大を行うときには、上記第1の方向と第2の方向の両方向の拡大フィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生し、上記第1の方向の画素順で順次入力した画素データ間をスキップするためのスキップ情報を生成し、上記第2の方向の入力画素列数と出力画素列数の拡大比を制御するためのタイミング情報を生成することを特徴とする請求項1記載の画像信号処理方法。

【請求項3】 フィルタ仕様情報に基づいて上記2次元画像の拡大または縮小の設定を行い、

上記2次元画像の拡大を行うときには、上記第1の方向と第2の方向の両方向の縮小フィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生し、上記所定のフィルタ演算処理がなされた後の上記第1の方向の各画素データをスキップするためのスキップ情報を生成し、上記第2の方向の入力画素列数と出力画素列数の縮小比を制御するためのタイミング情報を生成することを特徴とする請求項1記載の画像情報処理方法。

【請求項4】 第1の方向及び当該第1の方向と交差する第2の方向に配列する複数画素からなる2次元画像に対して、フィルタ演算を施す画像信号処理装置において、

フィルタ仕様情報に基づいて、上記第1の方向と第2の方向の両方向用のフィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生するプログラム及び係数発生手段と、

上記2次元画像を構成する上記第1の方向の画素順で順次シリアルに入力した画素データを、当該第1の方向の画素列毎にパラレルに出力するシリアル／パラレル変換手段と、

上記パラレルに変換された上記第1の方向の画素列の各画素データに対して、当該第1の方向用のフィルタプログラム及びフィルタ係数を用いた所定のフィルタ演算処理を施す第1のフィルタ演算手段と、

当該第1のフィルタ演算手段にて所定の演算処理がなされた上記第1の方向の画素列の各画素データに対して、必要に応じて更に上記第2の方向用のフィルタプログラム及びフィルタ係数を用いた所定のフィルタ演算処理を施す第2のフィルタ演算手段と、

上記第2のフィルタ演算手段から得られたパラレル画素データを、上記第2の方向の画素順に順次シリアル出力するパラレル／シリアル変換手段とを有することを特徴とする画像信号処理装置。

【請求項5】 上記第1の方向の画素順で順次入力した画素データ間をスキップするためのスキップ情報を生成するスキップ情報生成手段と、

上記第2の方向の入力画素列数と出力画素列数の拡大比を制御するためのタイミング情報を生成するタイミング情報生成手段とを設け、

上記2次元画像の拡大を行うとき、上記プログラム及び係数発生手段は、上記第1の方向と第2の方向の両方向の拡大フィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生することを特徴とする請求項4記載の画像信号処理装置。

【請求項6】 上記所定のフィルタ演算処理がなされた後の上記第1の方向の各画素データをスキップするためのスキップ情報を生成するスキップ情報生成手段と、

上記第2の方向の入力画素列数と出力画素列数の縮小比を制御するためのタイミング情報を生成するタイミング情報生成手段とを設け、

上記2次元画像の縮小を行うとき、上記プログラム及び係数発生手段は、上記第1の方向と第2の方向の両方向の縮小フィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生することを特徴とする請求項4記載の画像情報処理装置。

【請求項7】 上記第1の方向の画素順で順次入力した画素データまたは当該画素データ間をスキップするためのスキップ情報を生成するスキップ情報生成手段と、

上記第2の方向の入力画素列数と出力画素列数の拡大または縮小比を制御するためのタイミング情報を生成するタイミング情報生成手段とを設け、

上記2次元画像の拡大または縮小を行うとき、上記プログラム及び係数発生手段は、上記第1の方向と第2の方向の両方向の拡大または縮小フィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生することを特徴とする請求項4記載の画像信号処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、画素数変換処理や走査線変換等の画像信号処理を行う画像信号処理方法及び装置に関する。

【0002】

【従来の技術】近年になって、半導体技術、半導体の処理スピード性能の向上により、映像信号のデジタル信号処理が行われるようになってきた。また、最近では画像表示装置も従来のブラウン管に代わり、LCD (Liquid Crystal Display: 液晶ディスプレイ) 表示装置やプラズマディスプレイ装置等に代表される固定画素表示装置が広く普及しつつある。

【0003】また、最近では、いわゆるNTSC (National Television System Committee) 信号、PAL (Phase Alternation by Line) 信号などの標準テレビジョン放送方式のみならず、HDTV (High Definition Television) 信号や、VGA (Video Graphics Array) 信号、SVGA (Super VGA) 信号、XVGA (extended VGA) 信号など様々なフォーマットの信号を表示することが求められている。

【0004】これら様々なフォーマットでは、それぞれ扱う画素数がまちまちである。このようなそれぞれ画素数が異なる各種のフォーマットの映像信号を表示する場合、上記ブラウン管等のアナログ表示デバイスであれば、1走査線時間当たりの画素数に応じて電子ビームの偏向速度を変えてやれば済む。

【0005】しかし、上記固定画素表示装置においては、扱える画素数が固定しているため、上述のブラウン管の場合のような従来のアナログ技術は使えない。そのため、これら様々なフォーマットの信号を上述のような固定画素表示装置に対して表示させるためには、デジタル信号処理による任意の画素数変換、或いは走査線数変換が不可欠である。

【0006】ここで、上述したような画像の画素数変換を行うための技術として、従来より使用されてきた補間フィルタを用いた画素数変換の手法について、以下に説明する。なお、例えばデジタルフィルタによる補間フィルタの実現技術は、従来からデジタル信号処理分野の各種教科書に詳しく解説されているので、ここでは簡単に説明する。

【0007】先ず、画像の拡大や縮小と標本化周波数(画素数)の変換について説明する。

【0008】画像の拡大や縮小を実現するための演算操作と、画像の標本化周波数(画素数)を変換する演算操作(解像度の異なる画像規格間の変換)は、どちらも原画像の各画素位置に対して、最初存在しなかった位置の画素データを求める演算として考えることにより、全く同じ補間フィルタの演算操作でよいことがわかっていく。これを図11を用いて説明する。

【0009】図11の(a)には原画像のモデルを示す。この図11の(a)中の四角い枠は画面の一部分であるとする。また、図中丸印は画像の画素の位置を表すものとする。すなわち、この図11の(a)に示す部分原画像は、(横方向8画素)×(縦方向6画素)からなる。ここでは考え易くするために、扱いやすい少ない画素数にしている。この原画像を例えば10/7倍に拡大する場合を考える。なお、ここでの倍率は面積ではなく長さの比で表現するものとする。上記原画像を10/7に拡大するというのは、図11の(a)の画像(画素数)を拡大するのであって、表示画像規格は変わらず、画素の配列の仕方すなわち画素間隔などは図11の(a)と同じであり、その拡大結果は図11の(b)のようになる。すなわち、図11の(a)と図11の(b)とでは、10/7倍の拡大($10/7=1.429$ 倍)なので、図11の(b)の画像は図11の(a)の画像よりも1辺の長さが1.429倍に長くなり、図11の(b)の拡大画面を構成する画素数は1.429の自乗倍に増えている。

【0010】したがって、例えば水平方向で見た時は、同じ画像が原画像で8画素であったものが拡大後は、 $8 \times 10/7 = 11.429$ 画素になり、拡大後の各画素の画像に対する位置関係は違ってくる。なお、このため拡大後の各画素のデータ(輝度や色を表現するデータ)の値は、原画像のそれとは異なることになる。

【0011】図11の(d)は上記10/7倍の画像拡大の様子を水平方向についてのみ示している。図11の(d)中の矢印の先が各画素の位置を示していて、上側の R_i ($i=1, 2, \dots$) は原画素のデータであり、下側の Q_i ($i=1, 2, \dots$) は拡大後の補間画素のデータである。すなわち、 R_i の間隔は Q_i の間隔の10/7倍になっている。なお、この図11の(d)は水平方向の拡大の様子だけを示しているが、垂直方向についても図11の(d)と同様に考えることができる。ここでは垂直方向の説明は省略する。

【0012】拡大後の各画素のデータ(輝度や色を表現)の値は、その位置関係が原画像の各画素位置に対して図11の(d)のようになっていることから、画素1点1点について、ここで求めようとしている補間点の周辺のいくつかの原画像画素のデータの値を使って補間フィルタ演算、すなわち補間関数の畳み込み演算をすればよい。この補間演算については後述する。

【0013】次に、今度は画像の大きさをそのままにして、標本化周波数を例えば10/7倍にする場合を考えることにする。これは標本化周波数変換であるが、言い換えると、画像の大きさは同じで解像度が10/7倍高い画像規格に変換することと同じことであり、すなわち画素数を10/7倍にしていることと同じである。この場合は、原画像を図11の(a)とした時、その結果は図11の(c)のようになる。すなわち、1次元的には

画素数が1.429倍に増え、同じ面積内の画素数は1.429の自乗倍に増えている。

【0014】この画素数の多い画像規格に変換する演算操作は、先に説明した画像拡大の演算操作と全く同じでよい。それは、図11の(a)と図11の(b)の関係及び図11の(a)と図11の(c)の関係を見てわかるように、どちらも図11の(d)の位置関係からの補間フィルタ演算、すなわち補間関数の畳み込み演算を行えばよいからである。

【0015】次に、今度は画像の縮小の場合を説明する。図11の(a)と同様の図12の(a)を原画像のモデルとし、この原画像を例えば10/13倍に縮小する場合を考える。ここでは、原画像を縮小するのであって表示画像規格は変えないので、画素の配列の仕方すなわち画素間隔などは図12の(a)と同じであり、その結果は図12の(b)のようになる。つまり、10/13倍(10/13=0.769)倍の縮小なので、図12の(b)の画像は図12の(a)の画像の1辺の長さが0.769倍に縮み、この縮小画面を構成する画素数は0.769の自乗倍に減ることになる。

【0016】したがって、例えば水平方向に見た時は、同じ画像が原画像で8画素であったものが、縮小後に8×10/13=6.154画素になるのだから、当該縮小後の各画素の画像に対する位置関係は違ってくる。このため、縮小後の各画素のデータ(輝度や色を表現)の値は、原画像のそれとは異なることになる。

【0017】図12の(d)は上記10/13倍の画像縮小の様子を水平方向についてのみ示している。図中矢印の先が各画素の位置を示していて、上側の R_i ($i=1, 2, \dots$)は原画素のデータであり、下側の Q_i ($i=1, 2, \dots$)は縮小後の補間画素データである。すなわち R_i の間隔は Q_i の間隔の10/13倍になっている。なお、図12の(d)は水平方向の拡大の様子だけを示しているが、垂直方向についても図12の(d)と同様に考えることができる。

【0018】上記縮小後の各画素のデータ(輝度や色を表現)の値は、その位置関係が原画像の各画素位置に対して図12の(d)のようになっていることから、画素1点1点について、今求めようとしている補間点の周辺のいくつかの原画像画素のデータの値を使って補間フィルタ演算、すなわち補間関数の畳み込み演算をすればよい。この補間演算については後述する。

【0019】次に、今度は画像の大きさはそのままにして、標本化周波数を例えば10/13倍にする場合を考えることにする。これは標本化周波数変換であるが、言い換えると、画像の大きさは同じで解像度が10/13*

1. 最近傍近似法

2. 双一次近似法

3. キュービック(Cubic)近似法(図14の(d)、式(4))

などが知られている。なお、上記最近傍近似法の近似補

* 倍低い画像規格に変換することと同じことであり、画素数を10/13倍にすることと同じである。この場合は、原画像を図12の(a)とした時の結果は図12の(c)のようになる。すなわち1次的には画素数が0.769に縮み、同じ面積内の画素数は0.769の自乗倍に減っている。

【0020】この画素数の少ない画像規格に変換する演算操作は、先に説明した画像縮小の演算操作と全く同じでよい。それは、図12の(a)と図12の(b)の関係及び図12の(a)と図12の(c)の関係を見てわかるように、どちらも図12の(d)の位置関係からの補間フィルタ演算、すなわち補間関数の畳み込み演算をすればよいからである。

【0021】図11の(d)の場合も図12の(d)の場合も、最初原画像では存在しなかった位置の画素データ値を求める補間フィルタの演算操作が必要であり、結局、画像の拡大や縮小を実現するための演算操作も、画像の標本化周波数(画素数)を高く(多く)変換する演算操作も、低く(少なく)変換する演算操作も、どれも同じ演算操作でよいことになる。以下これらは「画素数変換」あるいは「補間フィルタ演算」と表現するものとする。

【0022】次に、原画像には存在しなかった位置の画素データ値を求める補間フィルタ演算、すなわち補間関数の畳み込み演算について、図13、図14及び式

(1)～式(4)を用いて説明する。

【0023】図13において横軸は図11の(d)や図12の(d)と同様に表しており、水平方向の位置関係を模式的に示している。また、図中の丸印は原画像の画素位置で、図中Sは原画像の標本化間隔、すなわち画素間隔を示している。そして、図中矢印hで指し示す位置を、補間すべき位置(補間点)とする。すなわち図中丸印の位置が図11の(d)或いは図12の(d)の前記 R_i であり、図中矢印hの位置が今注目している前記 Q_i の位置である。前記 Q_i の一つ一つについて、その Q_i を挟む周辺の R_i を使って畳み込み演算をして補間演算が実現される。

【0024】ここで、一般に、「標本化定理」によれば、理論的理想的な「補間」は図14の(a)及び次式(1)のようなsinc関数

$$f(x) = \text{sinc}(x) = \sin(x)/x \quad \dots (1)$$

を補間関数として、無限時間過去から無限時間将来まで畳み込み演算をすればよい。しかし実際にはそれは実現できず、現実的な課題としては、sinc関数をいかに有限期間の簡単な補間関数に近似するかという問題になる。そしてその簡単な近似法として、

(図14の(b)、式(2))

(図14の(c)、式(3))

(図14の(d)、式(4))

間関数を図14の(b)及び式(2)に示し、上記双一

次近似法の近似補間関数を図14の(c)及び式(3)に示し、上記キュービック(Cubic)近似法の近似補間関数を図14の(d)及び式(4)に示す。また、図1*

$$f(x) = 1 \quad (-0.5 < x \leq 0.5)$$

$$f(x) = 0 \quad (-0.5 \geq x, x < 0.5)$$

$$f(x) = 1 - |x| \quad (|x| \leq 1)$$

$$f(x) = 0 \quad (|x| > 1)$$

$$f(x) = |x|^3 - 2|x|^2 + 1 \quad (|x| \leq 1)$$

$$f(x) = -|x|^3 + 5|x|^2 - 8|x| + 4 \quad (1 < |x| \leq 2) \quad \dots (4)$$

$$f(x) = 0 \quad (2 < |x|)$$

補間後の1画素を演算するのに、原画像の画素データをそれぞれ、上記最近傍近似法では1画素、上記双一次近似法では2画素、上記キュービック近似法では4画素使って畳み込み演算をする。上記双一次近似法は加重平均をとっているのであって、線形補間として良く知られている。

【0026】ここで必要な畳み込み演算はいわゆるFIR(非巡回型)デジタルフィルタを用いればよく、近似補間関数の中心を補間点に合わせた時に、所定の画素数分だけ近傍の原画像の標本点で近似補間関数を標本化した値を補間フィルタ係数セットとして使う。

【0027】ここで注意すべきことは、今補間しようとしている各補間点ごとの、原画像画素位置に対する相対位置関係を示す前記図13の位相(ズレ)Pの違いにより、そのFIRフィルタの演算に必要な重み、すなわちフィルタ係数のセットが異なってくるということである。

【0028】例えば、上記双一次近似法で補間フィルタ演算をする場合であれば、 $P=0.0$ の場合は、その場合のフィルタ係数セットとなる2つの重み(フィルタ係数)は1.0と0.0となり、位置が一致する原画像の画素データ値をそのまま通すような係数セットとなるが、 $P=0.5$ の場合は上記2つの重み(フィルタ係数)は0.5と0.5の係数セットとなり、 $P=0.3$ の場合は上記2つの重み(フィルタ係数)0.7と0.3の係数セットとなる。同様に、キュービック近似法では、 $P=0.0$ の場合はその4つの重み(フィルタ係数)は0.0と1.0と0.0と0.0とになり、位置が一致する原画像画素のデータ値をそのまま通すような係数セットとなっているが、 $P=0.5$ の場合の4つの重み(フィルタ係数)は-0.125と0.625と0.625と-0.125の係数セットとなり、 $P=0.3$ の場合の4つの重み(フィルタ係数)は-0.063と0.847と0.363と-0.147との係数セットとなる。これらは上記式(2)～式(4)のxにPを代入して得られる。

【0029】次に、補間フィルタ演算、すなわち補間関数の畳み込み演算を実現するハードウェア構成について、その従来例を図15を用いて説明する。補間フィルタ演算はFIRデジタルフィルタにより行われ、ただ

*4の横軸は原画像の標本間隔で正規化されているものとする。

【0025】

... (2)

... (3)

しそのフィルタ係数が補間点ごとに変わらなければならない。したがって、図15のような構成の演算装置で積和演算すれば実現できる。なお、図15ではキュービック近似法を想定している。

【0030】図15は、各補間位置におけるデータの妥当な値を次々と演算(補間フィルタ演算)するFIRデジタルフィルタである。

【0031】この図15において、入力端子250には、水平走査された入力画像データ時系列が供給される。また、図中M1、M2、M3、M4として表す4つのレジスタ254～258は、入力端子250から供給された原画像の画素データ1つ分をそれぞれが記憶できるレジスタである。これらレジスタ254～258は直列接続されており、4段のシフトレジスタとして動作する。これらレジスタ254～258にて構成されるシフトレジスタには、上記水平走査された入力画像データ時系列がワード単位に入力され、これにより当該シフトレジスタは常に連続する4つの原画像画素データを記憶することになる。キュービック近似法は、補間点を挟む近傍左右2画素づつ計4画素の原画像データから補間点の妥当なデータ値を求める方法なので上記4段のシフトレジスタが必要になっている。

【0032】また、端子251には、制御されたクロック或いはクロックの制御信号Eが入力される。この制御信号Eは、上記4段のシフトレジスタをシフトさせたり、停止させたり制御する信号であり、図示しないコントローラから供給される。

【0033】端子252には、複数のフィルタ係数セット中から、求めたい補間点 Q_i の位相Pに対応すフィルタ係数セットを選ぶ選択信号Cが入力される。係数メモリ253は、先に説明した位相Pの種類ごとの複数のフィルタ係数セットを記憶するメモリであるため、上記選択信号はアドレスデータとなる。

【0034】乗算器259～262では、上記係数メモリ253からのフィルタ係数 $FC_1 \sim FC_4$ と上記4段のシフトレジスタの各レジスタ254～258からの各画素データとの乗算を行い、それぞれ得られた乗算結果を総和加算器263に出力する。

【0035】総和加算器263では、それぞれ供給された乗算結果の総和加算を行い、その加算結果が出力端子

264から補間フィルタ演算結果の出力画像データ時系列として出力される。

【0036】図15の構成は、従来よりハードウェアで実現されている。ここで、リアルタイム画像処理の場合においては、その標本化周期が50ナノ秒というようなオーダーの短時間であるために、リアルタイムの音の信号処理の場合などと違って、従来はソフトウェアプログラムによる実現例はほとんどなく、図15の各部分をパイプライン方式を多用したハードウェア専用化回路にして実現するのが一般的であった。

【0037】次に、上記図15のハードウェア構成において、画像をキュービック近似により例えば10/7倍に拡大する場合の動作を図16及び図17に示す表を使い説明する。10/7倍の画像拡大は図11で説明した通りであり、水平方向については図11の(d)の位置関係の各補間点ごとに図13のように考えて、図15のような装置で補間フィルタ演算をすればよい。図16に示す表はその場合における図15の各部の動作手順を示している。

【0038】図16に示す表はタイムチャートであり、表中のサイクルとは時刻すなわちクロックサイクルを意味している。表中のIN欄は入力データであり、図15の入力端子250に与えられる原画像画素データ値 R_i を示す。表中のC欄は図15の端子252に供給されるフィルタ位相選択信号Cであり、係数メモリ253のどの位相のフィルタ係数セットを使うかを選択し、そのサイクルで演算すべき Q_i のための位相 P_i を指定する。表中のE欄は図15の端子251に供給されるシフトレジスタ制御信号で、シフト(論理「H」)か停止(論理「L」)のどちらかの状態を指定する。この図16の表において、C欄とE欄に示す制御入力、予め何をするかによって図11の(d)のような検討を行い、所定の制御信号列を決めておき、図15の装置に与えることになる。なお、上記E欄の制御は次のクロックサイクルに対してその制御が影響されるものとする。表中の M_1, M_2, M_3, M_4 の各欄は図15の4段のシフトレジスタの各段のレジスタ254~258にそれぞれ記憶されている R_i を示す。表中のOUT欄は図15の出力端子264から、上記クロックサイクルで出力される出力データ Q_i を示す。なお、図15のようなハードウェア装置では、その乗算や総和演算の部分で何らかのパイプライン処理により、レイテンシー(高速演算実現のための本質的でない遅延)を生ずるのが普通であるが、ここではそのレイテンシーはないものとして考えることにする。

【0039】また、ここで前記係数メモリ253には、図17に示す表の8ビット表現係数の欄に示す10種類のフィルタ係数セットが記憶されているものとし、表中のC欄から与えられた位相 P_i に応じて、前記4つのフィルタ係数 FC_1, FC_2, FC_3, FC_4 を出力するものとする。今想定している10/7倍の画像拡大の場合に

はその補間演算に必要な位相の種類は図11の(d)のように10種類しかなくて、それが繰り返し使われるだけだからである。

【0040】図17の表中の10種類のフィルタ係数セットは図11の(d)の $Q_1 \sim Q_{10}$ についての10種類の位相 P に対応しているわけだが、これは図13で S を10等分した場合の位置についての10種類の P になっているので、図17の表では P の小さい順、すなわち0/10, 1/10, 2/10, ...の順で P_i の i の順を決めている。そして P_i を x として前記式(4)に代入して得た係数が浮動小数点表現係数であり、それを8ビットに語長制限して(ここでは最大振幅を128とした)8ビット表現係数を得ている。

【0041】上記図16の表及び図15の構成において、まず、第1サイクルではIN欄に示すように R_1 が現れる。この時、レジスタ254~258の M_1, M_2, M_3, M_4 には、 R_1 よりひとつ前のデータを R_{m0} 、そのもうひとつ前のデータを R_{m1} 、そのもうひとつ前を R_{m2} 、更にそのもうひとつ前を R_{m3} とした時、それぞれ $R_{m0}, R_{m1}, R_{m2}, R_{m3}$ が記憶されているものとする。

【0042】第1サイクルではE欄のシフト制御信号が論理「H」であり「シフト」を示しているので、次のクロックの立ち上がりでシフトレジスタはシフトし、第2サイクルでは M_1, M_2, M_3, M_4 としてそれぞれ $R_1, R_{m0}, R_{m1}, R_{m2}$ が記憶されている状態となる。第2サイクルでは、IN欄に示すように R_2 が与えられ、E欄のシフト制御信号は論理「H」すなわち「シフト」なので、次のクロックの立ち上がりでシフトレジスタはシフトして第3サイクルでは各レジスタはそれぞれ R_2, R_1, R_{m0}, R_{m1} となる。第3サイクルではIN欄から R_3 が与えられ、シフト制御信号は論理「H」なので、第4サイクルでは各レジスタはそれぞれ R_3, R_2, R_1, R_{m0} となる。

【0043】この第4サイクルではIN欄から R_4 が与えられ、C欄のフィルタ位相選択信号には図11の(d)における R_{m0}, R_1, R_2, R_3 に対する Q_1 の位置関係、すなわち位相 P_0 が与えられる。これは図13において $R_a = R_{m0}, R_b = R_1, R_c = R_2, R_d = R_3$ とする場合の Q_1 の位置関係を示す P_i が $i=0$ であるということである。よってC欄のフィルタ位相選択信号が今 P_0 なので、図15の係数メモリ253が出力する4つのフィルタ係数 FC_1, FC_2, FC_3, FC_4 は、図17の表中の位相 P_0 の場合のフィルタ係数セットすなわち0.0と1.0と0.0と0.0(8ビットの表現の場合0と128と0と0)が得られ、この係数を使って図15のような積和演算構成で畳み込みのための積和演算がなされて、出力端子264には R_1 の値がそのまま Q_1 となった出力データが得られる。

【0044】なお、この第4サイクルでは図16のE欄のシフト制御信号が論理「L」であり「停止」を示して

いるので、次の第5サイクルでもシフトレジスタの各レジスタ254~258は第4サイクルの時のデータを保持する。そして次の第5サイクルでは、図16の表中IN欄よりR₄が継続して与えられ、C欄のフィルタ位相選択信号には図11の(d)におけるR_{m0}, R₁, R₂, R₃に対するQ₂の位置関係、すなわち位相P₇が与えられているので、図17の表中位相欄のP₇に応じたフィルタ係数セットが前記係数メモリ253から出力されて、Q₂のデータ値を求める積和演算がなされ、出力端子264にはQ₂が得られる。ここで位相がP₇となっているのは、図11の(d)におけるR_{m0}, R₁, R₂, R₃に対するQ₂の位置関係に対応した、図13におけるR_a=R_{m0}, R_b=R₁, R_c=R₂, R_d=R₃の場合のPが、R₁とR₂の間隔Sに対して、R₁とQ₂の間隔が7/10だからである。

【0045】また、第4サイクルから第5サイクルになる時にシフトレジスタがシフトしなかったのは、図11の(d)においてQ₁とQ₂が、同じようにR₁とR₂の間に位置していて、補間フィルタ演算のための原画像信号の前後の4データの組合せが同じだからである。図16のE欄のシフト制御信号が停止すなわち論理「L」になる時はこのような場合であり、その場合次のサイクルのIN欄に与える入力データは、同じデータが連続される。

【0046】この第5サイクルではE欄のシフト制御信号が論理「H」であり、第6サイクルでは各レジスタはそれぞれR₄, R₃, R₂, R₁となる。そして第6サイクルではIN欄よりR₅が与えられ、C欄のフィルタ位相選択信号には図11の(d)におけるR₁, R₂, R₃, R₄に対するQ₃の位置関係、すなわち位相P₄が与えられているので、図17の表のP₄に応じたフィルタ係数セットが係数メモリから出力されて、Q₃のデータ値を求める積和演算がなされ、出力端子264にはQ₃が得られる。ここで位相がP₄となっているのは、図11の(d)におけるR₁, R₂, R₃, R₄に対するQ₃の位置関係に対応した、図13におけるR_a=R₁, R_b=R₂, R_c=R₃, R_d=R₄の場合のPが、R₂とR₃の間隔Sに対して、R₂とQ₃の間隔が4/10だからである。これは、Q₂の時のP₇の7/10に対して、更に7/10が累加されて14/10となり、このうち10/10が原画像データ1個分に相当するので減算されたためである。

【0047】すなわち、Q_iのiが1増えるごとにPは0.7ずつ増えており、しかし整数分はデータ遅延として扱えるのでモジュロ演算されるのである。このPの増分をここでP_dと定義することにする。

【0048】以下、同様にして図15の構成の上で、図11の(d)によって決まる図16の表の手順と制御により、順にQ₁が求められ、補フィルタ間演算が実現される。

【0049】なお、図15は補間フィルタ演算の実現であって、ここまでの説明では正確にはデータレートの変換は実現されていない。すなわち画像の画素数変換などを施した時にその入出力のデータレートは必然的に違ってくるが、入出力データはそれぞれ画像データとして定常的な一定速度のデータ流になっていて欲しいのが普通である。しかし、例えば図16の表の画素数が増える変換の例では、出力は一定になっているものの、入力データ列は走ったり止まったりする必要が生じている。画素数が減る変換の場合ならこれは逆に入力是一定になるものの、出力データ列が出力されたり出力されなかったすることとなる。この問題については、データの流れ速度の問題であり、これは図15のような構成に前後どちらかに、そのデータ流を一定化させたり、一定のデータ速度を間欠化するためのバッファメモリ、或いはFIFOメモリを配置することで容易に実現される。このためのバッファメモリの構成などについてはメモリ制御の問題であり、説明を省略する。

【0050】

20 【発明が解決しようとする課題】 上述したように、従来の画素数変換は、前述の図15に示したような固定された積和演算回路からなる構成にて行われている。

【0051】しかし、このような固定された積和演算回路の場合、例えばタップ数や変換比率を変更したい時に回路設計からやり直さなければならず、手間と時間がかかるという問題点があった。また、複数の変換比率にて画素数変換を行う必要があるような場合には、それぞれの変換比率に対応する画素数変換装置を複数用意しなければならないという問題点もあった。

30 【0052】そこで、本発明はこのような状況に鑑みてなされたものであり、例えばタップ数や変換比率を変更したい時に、容易かつ短時間でそれらの変更が可能であり、また、複数の変換比率にて画素数変換を行う必要がある場合でもそれぞれの変換比率に対応する複数用意する必要が無い画像信号処理方法及び装置を提供することを目的とする。

【0053】

40 【課題を解決するための手段】 本発明の画像信号処理方法及び装置は、第1の方向及びそれと交差する第2の方向に配列する複数画素からなる2次元画像に対して、フィルタ演算を施す画像信号処理方法及び装置であり、フィルタ仕様情報に基づいて第1、第2の両方向用のフィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生し、第1の方向の画素順で順次シリアルに入力した画素データをパラレルに変換し、このパラレル変換された第1の方向の画素列の各画素データに対して第1の方向用のフィルタプログラム及びフィルタ係数を用いた所定のフィルタ演算処理を施し、その後の第1の方向の画素列の各画素データに対して、必要に応じて更に第2の方向用のフィルタプログラム及びフィルタ係数を用い

た所定のフィルタ演算処理を施し、得られたパラレル画素データを第2の方向の画素順に順次シリアル出力することにより、上述した課題を解決する。

【0054】ここで、本発明の画像信号処理方法及び装置では、2次元画像を拡大または縮小するとき、その拡大または縮小の設定に応じた第1、第2の両方向の拡大または縮小フィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生し、第1の方向の画素順で順次入力した画素データまたは当該画素データ間をスキップするためのスキップ情報を生成し、第2の方向の入力画素列数と出力画素列数の拡大または縮小比を制御するためのタイミング情報を生成する。

【0055】すなわち、本発明によれば、フィルタプログラム及びフィルタ係数をフィルタ仕様情報に基づいて自動的に設定し、この自動的に設定されたフィルタプログラム及びフィルタ係数に基づいてフィルタ演算を行っている。また、本発明によれば、2次元画像を第1の方向と第2の方向に分けて考え、これら第1の方向と第2の方向のそれぞれに対して分離してフィルタ演算を行うようにしているため、フィルタ演算のためのフィルタプログラム及びフィルタ係数を容易かつ効率的に設計できるようにしている。さらに、このようにフィルタ演算を分離することで、第1の方向と第2の方向のフィルタ演算の独立性や保守性を高めるようにしている。

【0056】

【発明の実施の形態】以下、本発明の好ましい実施の形態について、図面を参照しながら説明する。

【0057】先ず、画素数変換処理のための補間フィルタ演算を前述の図15のようなハードウェア構成ではなく、ソフトウェアプログラム処理にて実現する手法について説明する。

【0058】当該ソフトウェアプログラムにて補間フィルタ演算を行う構成としては、例えば図1に示すような画像用のいわゆるSIMD(Single Instruction Multiple Data Stream)制御されるリニアアレイ型多並列プロセッサ120が知られている。このプロセッサ120は、いわゆるDSP(digital signal processor)にて実現されている。このプロセッサ120は大きく分けて、入力SAM(serial access memory)部121と、データメモリ部122と、ALU(arithmetic and logical unit)アレイ部123、出力SAM部124、プログラム制御部125とを有してなるものである。

【0059】上記入力SAM部121、データメモリ部122、ALUアレイ部123、出力SAM部124は、全体でリニアアレイ(直線配列)型に多数並列化された要素プロセッサ群を構成しており、この多数の要素プロセッサは、プログラム制御部125内にある共通の一つのプログラム制御部により連動して制御(SIMD制御)される。プログラム制御部125には、プログラムメモリとそのプログラムの歩進のためのシーケンス制

御回路などがあり、プログラムメモリに予め書き込まれたプログラムに従って各部分に接続されている各種制御信号を発生して各部分を制御する。

【0060】なお、入力SAM部121、データメモリ部122、出力SAM部124は基本的にメモリであり、これらのメモリのための「ロウ(ROW)」アドレスデコードは、詳細に説明しないが図1においては、プログラム制御部125に含まれているものとして以下説明する。

【0061】多数並列化された要素プロセッサの単一エレメント分は、図1中に斜線で示したような縦の細長い範囲であり、これが図で横方向に直線配列で並んでいる。すなわち一般にプロセッサと呼ばれるために必要な図2のような一般的な構成要件を、図1の斜線で示した縦の細長い要素プロセッサはそれぞれが満たしている。図2の入力バッファメモリ(IQ)130に相当するのが図1の入力SAM部121である。図2の出力バッファメモリ(OQ)131に相当するのが図1の出力SAM部124である。図2のデータメモリ(RF)132に相当するのが図1のデータメモリ部122である。データメモリ132のデータを、必要に応じて選んで読み出して演算をするためのALU133に相当するのが図1のALUアレイ部123である。

【0062】なお、図2の構成では、入力端子DINに供給された入力データは一旦入力バッファメモリ(IQ)130に入力され、そこから記憶のためのデータメモリ(RF)132に必要な分ずつ移される。そのデータメモリ132に今記憶された最新のデータや、過去に記憶したデータや演算途中のデータなどを必要に応じて選んで読み出してはALU133で演算し、再びデータメモリ(RF)132に書き込むという作業をプログラム制御で繰り返し、その結果得られた演算結果をバッファメモリ(OQ)131に移動し、所定の速度やフォーマットで出力する。図2の構成は、様々な用途に利用できる汎用なプロセッサ形態になっている。

【0063】普通のプロセッサではそのハードウェアは一般にワード処理プロセッサであり、ワードを単位として処理するが、図1の斜線で示した縦の細長い範囲で示す1つの要素プロセッサにおいて、入力SAM部121、データメモリ部122、出力SAM部124はメモリの「カラム」になっており、またALUアレイ部123は1ビットALUであり、事実上フルアダー(全加算器)を主体にした回路になっている。そのため普通のプロセッサとは違ってビット処理プロセッサであり、ビットを単位として処理する。普通のCPU(中央処理装置)で言う8ビットマシンとか16ビットマシンという言い方に対応させれば1ビットマシンである。ビット処理プロセッサはハードウェアが小さく、普通には実現できない程多数の並列数を実現できるので、画像用の場合、要素プロセッサ直線配列の並列数は、映像信号の一

水平走査期間の画素数(H)に一致させている。

【0064】更に、この要素プロセッサの構造は一例として図3のように書くことができる。ただし図3の各セルの構造は、理解を容易にするために非常に一般的なものとしてあるし、また同じ回路が複数並ぶ部分はその要素回路で代表して書いてある。

【0065】図1の入力SAM部121の一つの要素プロセッサ分は、入力ポインタ121Aからの制御を受ける縦に並んだ複数の入力SAMセル121Bである。入力データバス及び入力SAMセル121Bは図1の入力信号のビット数分(ISB)だけ縦に並べて用意されるのだが、図3ではそれを省略して一つのセル分の図で代表して表示している。図1のデータメモリ部122の一つの要素プロセッサ分は、図3におけるメモリセル122Aを、図1のデータメモリ部122の容量のビット数分(MB)用意して縦に並べているのだが、図3ではそれを省略して一つのセルだけ代表して表示している。上記ビット数分(MB)は演算処理の作業メモリとして必要なだけ用意される。前記データメモリ部122のメモリセル122Aは、2本の読み出しビット線と1本の書き込みビット線を持った3ポートメモリになっており、読み出しアドレス2組と書き込みアドレス1組を同時に与えることができるようになっている。前記ALUアレイ部123の一つの要素プロセッサ分は図3におけるALUセル123Aである。ここでALUセル123A中の正味のALU123B部分は1ビットALUであり、全加算器(フルアダー)程度の回路である、ALUセル123A中には他に、ALUの入力選択などのためのセレクト(SEL)回路等がある。なおALUセル123A中でFFにて示す部分はフリップフロップ(1ビットレジスタ)である。前記出力SAM部124の一つの要素プロセッサ分は、出力ポインタ124Aからの制御を受ける縦に並んだ複数の出力SAMセル124Bである。出力データバス及び出力SAMセル124Bは図1の出力信号ビット数分(OSB)だけ縦に並べて用意されるのだが、図3ではそれを省略して一つのセル分の図で代表して表示している。

【0066】また、この図3の入力SAM読み出し信号S_{IR}、メモリ書き込みアクセス信号S_{WA}、メモリ読み出しアクセス信号S_{RAA}、S_{RBA}、出力SAM書き込み信号S_{OW}などはメモリセルのワード線であり、セルを横に通過すると共に、横方向に並ぶ同じ回路要素を同様に接続しながら通過する。これらメモリセルのワード線は、図1のプログラム制御部125内でアドレスデコードがされているものとする。

【0067】さらに、図3において、セルを縦に通過する接続線すなわちビット線とポインタ信号線は、縦方向に並ぶ回路要素を同様に接続しながら通過するものとする。入力データバスは、それぞれ横に並ぶ同じ回路要素、すなわち入力SAMセル121Bを同様に接続しな

がら通過している。出力データバスもそれぞれ横に並ぶ同じ回路要素、すなわち出力SAMセル124Bを同様に接続しながら通過している。

【0068】次にこのプロセッサ120の動作を図1、図3を使って説明する。

【0069】入力端子D_{IN}に与えられた入力映像信号データは入力データバスを介して入力SAM部121に導かれる。入力ポインタ121Aは一つの要素プロセッサにだけ論理「H」を立てた1ビット信号すなわち入力ポインタ信号S_{IP}を発生し、論理「H」で指定された要素プロセッサの入力SAMセル121Bに入力映像信号データが書き込まれる。ポインタで指定された入力SAMセル121Bでは、トランジスタT_{r1}がONし、コンデンサC₁が入力映像信号データに応じた電位になる。

【0070】入力ポインタ信号S_{IP}の入力指定は、入力映像信号のクロック毎に左端の要素プロセッサから右端の要素プロセッサに順に移動するので、入力映像信号データは左端の要素プロセッサの入力SAMセル121Bから順に右方向の要素プロセッサのSAMセルに記憶場所が移動し、横に並んだ要素プロセッサ数が入力映像信号の一水平走査期間の画素数分(H)以上あるので、入力映像信号の一水平走査期間分を入力SAM部121に蓄積することができる。このような入力動作は、水平走査期間毎に繰り返される。

【0071】プログラム制御部125は、このようにして入力映像信号の一水平走査期間のデータが入力SAM部121に蓄積される毎に、プログラムに従って入力SAM部121、データメモリ部122、ALUアレイ部123、出力SAM部124を以下のようにSIMD制御して処理の実行をする。このプログラム制御は水平走査期間ごとに繰り返される。水平走査期間の時間をこのプロセッサの命令サイクル周期で割り算したステップ数だけのプログラムが組める。SIMD制御であるから、以下の動作は全ての要素プロセッサにおいて同時に同様に実行される。

【0072】一水平走査期間分の入力SAM部121に蓄積された入力映像信号データは、次の一水平走査期間において、必要に応じて入力SAM部121からデータメモリ部122へ移され、その後の演算処理に使われる。

【0073】最初の入力SAM部121からデータメモリ部122への転送動作は、入力SAM部121の必要なビットの記憶内容を入力SAM読み出し信号S_{IR}により選んでアクセスしては、転送先のデータメモリ部122の所定のメモリセルへメモリアクセス信号S_{WA}を出して書き込んでいくことにより実現する。入力SAM読み出し信号S_{IR}により選ばれた入力SAMセル121Bでは、トランジスタT_{r2}がONとなって、コンデンサC₁の電位に応じた転送データ信号が縦の書き込み用ビット線に生じる。この時、信号S_{BC}がイネーブルされ、メモ

リアクセス信号 S_{WA} で選択されたデータメモリ部122の所定のメモリセルのトランジスタ Tr_{11} がONとなつて、コンデンサ C_{11} が転送データに応じた電位になる。このデータ転送は縦方向の書き込み用ビット線を経由して1サイクルに1ビットづつ行われる。入力SAM部121の各入力SAMセル121Bからの読み出し信号 S_{IR} とデータメモリ部122の各メモリセル122Aへのメモリアクセス信号 S_{WA} は同じアドレス空間にあり、しかしそれぞれのロウデコーダでデコードされてワード線として与えられている。

【0074】データメモリ部122には、過去に上述のようにして書き込まれた入力データや演算途中のデータが記憶されている。それらのデータ或いはALUセル123A中の1ビットレジスタ(F F)に記憶したデータを用いて、ALUでのビット単位の必要な演算処理を順次進めることができる。例えばデータメモリ部122のあるビットのメモリセル122Aのデータと別のビットのメモリセル122Aのデータを加算して再びデータメモリ部122の更に別のビットのメモリセル122Aに加算結果を書き込む場合は次のようになる。データメモリ部122のあるビットのメモリセル122Aへ読み出し信号 S_{RAA} を出して、そのセルのトランジスタ Tr_{13} をONにして、コンデンサ C_{11} に記憶されていたデータを片方の読み出し用ビット線に出す。同時にもう一つのビットのメモリセル122Aへも読み出し信号 S_{RBA} を出して、そのセルのトランジスタ Tr_{14} をONにして、コンデンサ C_{11} に記憶されていたデータをもう片方の読み出し用ビット線に出す。これら2つの読み出されたデータは、ALUアレイ部123のセクタ(SEL)に所定の経路選択をさせ、ALU123Bに加算をさせ、ALU123Bからの出力は、信号 S_{BCA} をイネーブルして演算結果データを書き込み用ビット線に出し、そしてデータメモリ部122の所定の書き込みメモリセル122Aへ書き込みメモリアクセス信号 S_{WA} を出してそのセルのトランジスタ Tr_{11} をONにして、コンデンサ C_{11} がALU出力データに応じた電位になる。このALUセル123Aでの演算動作は、ALU制御信号 $S_{ALU-CONT}$ によりプログラムから指定される。ALUセル123Aで演算した結果は、再びデータメモリ部122に書き込むか、或いは必要に応じてALUセル123A中の1ビットレジスタ(F F)に記憶することもできる。加算の場合は普通、キャリーをビットレジスタ(F F)に、サムをデータメモリ部121に導く。

【0075】このようにしてデータメモリ部122から、プログラムに応じてデータを読み出してはALUアレイ部123で必要な算術演算或いは論理演算を施し、再びデータメモリ部122の所定のアドレスに書き込むことが出来る。この演算処理は全てビット処理であり、サイクル当たり1ビットづつ処理を進める。例えば論理演算を実行するには、それが8ビットのデータ同士であ

れば、1ビットづつ処理を進めるので必然的に8サイクルかかることになる。また8ビットのデータ同士の加算であれば、必然的に9サイクルかかることになる。8ビットのデータ同士の乗算の場合は、64回のビット加算と等価なので、単純には64サイクル強かけて実現されることになる。

【0076】一水平走査期間の時間内に処理すべき演算処理が済むと、その水平走査期間のうちに、プログラムの最後の部分でその水平走査期間分の演算処理済みの出力データを出力SAM部124に移す必要がある。データメモリ部122の出力すべきデータが記憶されている所定のメモリセル122Aへメモリアクセス信号 S_{RAA} 或いは S_{RBA} を出してそのセルのトランジスタ Tr_{13} 或いはトランジスタ Tr_{14} をONにして、コンデンサ C_{11} に記憶されていたデータを読み出し用ビット線に出す。ALUアレイ部123には通過する制御信号を出し、出力SAM部124には所定のビットの出力SAMセル124Bにデータ転送されるようにその出力SAMセル124Bに書き込み信号 S_{OW} が出され、そのセルのトランジスタ Tr_{17} がONして、コンデンサ C_4 がそのデータに応じた電位となる。データは縦方向のビット線を経由して1ビット1ビットデータ転送される。この時データ移動に際してALUで何らかの処理をしても良いし、しなくてもよい。出力SAM部124の各出力SAMセル124Bへの書き込み信号 S_{OW} とデータメモリ部122の各メモリセル122Aへのメモリアクセス信号は同じアドレス空間にあり、しかしそれぞれのロウデコーダでデコードされてワード線として与えられている。

【0077】以上のように、一水平走査期間の時間のうちに、入力SAM部121に蓄積された入力データのデータメモリ部122への移動、必要な演算処理、そして出力SAM部124への出力データの移動までが、ビットを単位とするSIMD制御プログラムで制御され実行される。このプログラム処理は水平走査期間を単位として繰り返される。SIMD制御なので、全ての要素プロセッサは連動動作をしており、水平走査期間分の画素数(H)に対して、同じ処理がなされる。

【0078】このような水平走査期間単位のプログラム処理が終わって出力SAM部124に移された出力データは、更に次の水平走査期間に次のように出力SAM部124から出力される。出力映像信号データは出力SAM部124から出力データバスを介して出力端子DOUTへ導かれる。

【0079】出力ポインタ124Aは一つの要素プロセッサにだけ論理「H」を立てた1ビット信号すなわち出力ポインタ信号 S_{OP} を発生し、その論理「H」で指定された要素プロセッサの出力SAMセル124Bから出力データが出力データバスに読み出され、出力映像信号データとなる。当該ポインタで指定された出力SAMセル124Aではトランジスタ Tr_8 がONとなって、コン

デンサC₄の電位に応じた出力信号が出力データバスに得られる。出力ポインタ信号の論理「H」が出力映像信号のクロックごとに左端の要素プロセッサから右端の要素プロセッサに移動することにより、出力データの読み出しは左端の要素プロセッサの出力SAMセル124Bから順に右方向の要素プロセッサのSAMセルに移って行き、横に並んだ要素プロセッサ数が映像信号の一水平走査期間の画素数分(H)以上であるので、出力映像信号の一水平走査期間分のデータを出力SAM部124から出力できる。このような出力動作は水平走査期間毎に繰り返される。

【0080】ここで、

1. 入力データの入力SAM部121への書き込みによる入力動作。

【0081】2. プログラム制御部125のSIMD制御による、入力SAM部121に蓄積された入力データのデータメモリ部122への移動、必要な演算処理、出力SAM部124への出力データの移動。

【0082】3. 出力データの出力SAM部124からの読み出しによる出力動作。

【0083】の3つの動作は、映像信号の一水平走査期間を単位とするパイプライン動作になっており、一つの水平走査期間の入力データに注目すれば、それぞれの動作は一水平走査期間の時間づつずれた形で実行されるが、3つの動作は連続して同時に並行して進行できる。

【0084】このような画像用リニアアレイ型多並列プロセッサにおいては、水平方向の同じ画素位置では、どの水平走査期間についても一つの要素プロセッサがいつも担当するので、過去の水平走査期間の入力データを後の水平走査期間までデータメモリ部122内に保存するために、データを入力SAM部121からデータメモリ部122に移す際にそのアドレスを水平走査期間毎にずらすメモリアドレス発生の工夫をすることにより、それぞれの要素プロセッサでは、縦方向の所定の連続数の画素データをデータメモリ部122内に持つことができ、垂直方向のFIRデジタルフィルタ演算が実現できることが知られている。

【0085】また図1や図3の説明で触れなかったが、要素プロセッサ列において、隣接する数プロセッサ間ではプロセッサ間通信のための接続が用意されている。具体的には図1のデータメモリ部122とALUアレイ部123の間にその接続面線があり、図3のALUセル123Aのメモリセル122A側のビット線がALUセル123A内のセクタSELに入出力されるところで、自分の要素プロセッサ内のデータメモリ部122のビット線だけでなく、左右の隣接する数プロセッサのデータメモリ部122のビット線の中からビット線接続を選ぶようにしているので、これを利用して各要素プロセッサは左右いくつかの隣接する要素プロセッサのデータメモリ部122をアクセスすることができる。これによ

て、水平方向に所定の連続数の画素データを扱うことができ、水平方向のFIRデジタルフィルタ演算も実現できるようになっている。

【0086】ただしこの左右いくつかの隣接する要素プロセッサのデータメモリ部122のアクセスは、SIMD制御であるから、例えば右隣の要素プロセッサのデータメモリ部122をアクセスするサイクルでは、全ての要素プロセッサは右隣の要素プロセッサのデータメモリ部122をアクセスしていることになる。もちろんこれは普通のFIRデジタルフィルタ実現には支障がない。また、直接プロセッサ間通信接続のない少し離れた要素プロセッサのデータアクセスについては、近傍のプロセッサ間通信を繰り返すことにより、多少プログラムステップは増えるが実現可能である。

【0087】上述したように、画素数変換処理のための補間フィルタ演算は、SIMD制御されるリニアアレイ型多並列プロセッサによるソフトウェアプログラム処理にて実現可能である。

【0088】ここで、前述した図1の本発明実施例の画像信号処理装置の構成は、さらに詳細には図4に示すように構成されるものである。

【0089】すなわち、本実施例の画像信号処理装置は、リニアアレイ型多並列プロセッサを用いて画像処理を行う際、水平及び垂直フィルタを2つのプロセッサエレメント群で独立に実行するようにしている。このように、分離型フィルタ構成とすることで、効率的な水平、垂直フィルタを一般の1次元フィルタ設計手法により実現できる。また、これを2つの独立したプロセッサエレメント群で実行するという構成により、プログラムの独立性、保守性などを高めることができ、利用し易くしている。

【0090】前記図1をさらに詳細に示す図4のリニアアレイ型多並列プロセッサ20は、複数の要素プロセッサ(プロセッサエレメント:PE)23を有してなり、各要素プロセッサ23は、上下2組のプロセッサエレメント群21, 22を形成している。プロセッサエレメント群21, 22の要素プロセッサ23の数は、少なくとも画像の水平1ライン分を構成する画素の数より多く、プロセッサエレメント群21は例えば水平用として、またプロセッサエレメント群22は例えば垂直用として設けられている。これらプロセッサエレメント群21, 22の各要素プロセッサ23は、それぞれが前記図1のデータメモリ部122及びALUアレイ部123の機能を有するものである。

【0091】また、コントローラ26, 28と係数メモリ27, 29は前記図1のプログラム制御部125と対応している。

【0092】コントローラ26は、プロセッサエレメント群21と係数メモリ27を前述したようなSIMD制御するものであり、係数メモリ27は演算に必要なデー

タをプロセッサエレメント群21に供給する働きを持つ。プロセッサエレメント群21に含まれる全ての要素プロセッサ23は、コントローラ26の制御信号に基づく同一の演算処理を行う。

【0093】また、コントローラ28、係数メモリ29についても同様の関係がある。すなわち、コントローラ28は、プロセッサエレメント群22と係数メモリ29をSIMD制御するものであり、係数メモリ29は演算に必要なデータをプロセッサエレメント群22に供給する働きを持つ。プロセッサエレメント群22に含まれる全ての要素プロセッサ23は、コントローラ28の制御信号に基づく同一の演算処理を行う。

【0094】このようにコントローラ26、28が2系統設けられているので、プロセッサエレメント群21、22はそれぞれ独立に動作できる。

【0095】入力レジスタ24は、図1の入力SAM部121と対応し、シリアル→パラレル変換を行うメモリである。この入力レジスタ24では、画像データが1画素ずつ順次入力され、1ライン分揃った時点で、並列に読み出される。当該入力レジスタ24への画素データ入力の際には、例えば隙間なく詰め込むのではなく、任意の場所にゼロを挿入して全体の画素数を拡大するアップサンプラーとしても動作可能となっている。

【0096】出力レジスタ25は、図1の出力SAM部124と対応しており、パラレル→シリアル変換を行うメモリである。この出力レジスタ25では、1ライン分の画素データを並列に入力し、それを1画素ずつ順次出力する。当該出力レジスタ25からの画素データ出力の際には、例えば任意の場所の画素データをスキップして全体の画素数を縮小するダウンサンプラーとしても動作可能となっている。

【0097】次に、この図4に示したリニアアレイ型多並列プロセッサ20の動作について説明する。

【0098】図4において、画像入力端子10から水平方向に1ライン分の画素データが、入力レジスタ24に順次入力された後、並列に上側のプロセッサエレメント群21に供給される。プロセッサエレメント群21では、各要素プロセッサ23が受け取った各々の画素データに対して、コントローラ26からの制御命令と係数メモリ27からの係数データを用いてフィルタ演算処理を行う。上側のプロセッサエレメント群21による処理結果は、そのまま下側のプロセッサエレメント群22に並列に入力され、必要であれば、更にフィルタ演算処理が施される。下側のプロセッサエレメント群22による処理結果は、出力レジスタ25に並列に入力され、1画素ずつ順次出力されて1ライン分の処理結果が得られる。

【0099】以上により、リニアアレイ型多並列プロセッサ20は、水平方向に1ラインを形成する画素を一度に処理できる。これを垂直方向のライン数だけ繰り返すと、2次元の画像処理ができる。当該プロセッサ20の

具体的動作については前述したように既に説明してあるので、ここではその詳細についての説明は割愛する。

【0100】次に、上記プロセッサエレメント群21、22によるフィルタ演算処理の内容について述べる。なお、リニアアレイ型多並列プロセッサ20を用いて水平フィルタ、垂直フィルタを実現する方法に関しても本件出願人は既に開示しているので、ここでは簡単に説明する。

【0101】プロセッサエレメント群21、22を構成する各要素プロセッサ23は、互いに左右の近傍の要素プロセッサ23と通信を行うことができ、このため、同一ライン上の複数の画素データを用いた演算ができる。また、要素プロセッサ23は、前述したように、必要に応じて複数ライン分の画素データを保持するメモリを備えているので、ライン間に跨る演算も行える。

【0102】ここで、上記画素数変換として、例えば水平、垂直方向ともに2/3倍に画素数を縮小する場合を例に挙げると、当該2/3倍縮小画素数変換は、例えば図5の(A)に示すように、水平方向9画素で垂直方向6画素の画像から、例えば図5の(B)に示すように、水平方向6画素で垂直方向4画素の画像を生成するようなことである。このように図5の(A)の画像から図5の(B)画素を生成するためには、例えば図5の(C)及び図中の式(a)のように図5の(A)中の4点の原画素D1~D4から補間画素Yを生成するような処理を、当該図5の(A)の画像中の各画素について行い、得られた補間画素Yから図5の(B)の画像を生成する。

【0103】この図5の画素数変換の例では、2ライン分の入力画素データが揃えば、出力画素が演算できる。また、水平、垂直方向に2/3倍の画素数変換が行われるが、水平方向に関しては出力レジスタ25のスキップ機能を用いて、画素数を2/3に間引く。一方、垂直方向に関しては、「2ライン入力して1ライン出力する」、「1ライン入力して1ライン出力する」を交互に繰り返すことで、出力ライン数は入力ライン数の2/3になる。

【0104】ここで、2次元フィルタのうち、上述のような水平、垂直の分離型フィルタは、2つの1次元フィルタにて実現することができる。この分離型フィルタには、必要な演算器の数を減らせることや、多くの1次元フィルタ設計手法が利用できることなどの利点があり、実用上極めて重要である。簡単に説明すると、例えば図5のような画素数変換を行うフィルタ($Y=C_1D_1+C_2D_2+C_3D_3+C_4D_4$)は、($C_1=C_5C_7$, $C_2=C_6C_7$, $C_3=C_5C_8$, $C_4=C_6C_8$)となるようなフィルタ係数C5~C8が存在するとき、($H_1=C_5D_1+C_6D_2$, $H_2=C_5D_3+C_6D_4$)なる水平フィルタと、($Y=C_7H_1+C_8H_2$)なる垂直フィルタに分離できる。

【0105】ここで、上述したような水平、垂直フィル

タを2つのプロセッサエレメント群21, 22にて独立に実行するためのプログラムや係数、画素数変換のための制御情報等は、図6に示すような流れにより自動的に生成することができる。すなわち、図6に示す流れにより上記プログラムや係数等を自動的に発生すれば、例えば、フィルタの仕様を決定するだけで、そのフィルタ仕様に基づく画素数変換装置が実現可能となり、図4の構成では、図6の流れにて発生されたプログラムや係数を入れ換えるだけで、回路の変更をすることなく多種の画素数変換を含むフィルタ処理を行えるようになる。

【0106】この図6において、先ず、所望のフィルタを設計するための例えばタップ数や変換比率を設定するためのフィルタ仕様30に基づき、フィルタ設計回路33が水平及び垂直のフィルタプログラム34, 36及びフィルタ係数(フィルタ係数セット)35, 37を発生する。上記水平フィルタプログラム34は端子41及び図4の端子11を介して前記コントローラ26に送られ、上記水平フィルタ係数35は端子42及び図4の端子12を介して前記係数メモリ27に送られる。また、上記垂直フィルタプログラム36は端子43及び図4の端子13を介して前記コントローラ28に送られ、上記垂直フィルタ係数37は端子44及び図4の端子14を介して前記係数メモリ29に送られる。フィルタ係数セット35, 37は、一般の1次元フィルタ設計ツールを用いても、また、ライブラリから選択してもよい。画像の拡大縮小で、画素数の変化を伴う場合は、プロセッサエレメント割当回路31がデータ入力レジスタ24及びデータ出力レジスタ25に対するプロセッサエレメントスキップ情報32を発生し、水平方向の画素数を調整する。上記スキップ情報32は、端子45を介して図4の端子15に送られ、入力レジスタ24或いは出力レジスタ25に送られることになる。また、垂直方向に関しては、入力ライン数と出力ライン数の比を制御するために、タイミング設計回路38がタイミング情報39を発生し、これが垂直フィルタプログラム36に組み込まれる。なお、図4では、上下のプロセッサエレメント群21, 22にそれぞれ水平、垂直フィルタを割り当てているが、逆の割り当てを行っても、同一の処理結果が得られる。

【0107】次に、上記プロセッサエレメント割当回路31の具体的な動作について説明する。

【0108】当該プロセッサエレメント割当回路31は、水平方向の画素数変換のためのスキップ情報32を生成する。スキップ情報32は、水平フィルタプログラム34において、フィルタ仕様30にて設定される変換比率に基づき、水平画素数を変換するということに作用する。すなわち入出力の画素数の比を $N_i : N_o$ (N_i, N_o は互いに素)とすると、変換比率 $R = N_o / N_i$ によって拡大と縮小に動作が分けられる。

【0109】先ず、 $R > 1$ (拡大) の場合を説明する。

【0110】当該 $R > 1$ の場合、入力レジスタ24に設定するスキップ情報としては、例えば左から i 番目の入力レジスタに対し、 $(i \bmod N_i) < N_o$ であれば通常動作し、そうでなければスキップ動作するというようなスキップ情報を設定する。ただし、 $(i \bmod N_i)$ は、 i を N_i で割った余りを表す。例えば、 $3/2$ 倍の拡大を行うときは、入力レジスタ24に対し、次のようなスキップ情報32が送られる(ただし、○は通常動作を、×はスキップ動作を表す)。

10 【0111】○○×○○×○○×・・・

ここに、画像入力端子10から、例えば

a b c d e f g h i・・・

なるデータが入力された場合、入力レジスタ24には
a b 0 c d 0 e f 0 g h 0 i・・・

とついに格納され、データの個数が $3/2$ 倍に増えた形でプロセッサエレメント群21に渡される。この入力レジスタ24に格納されるデータのゼロが挿入された位置には、後に水平フィルタプログラム34によって補間画素が補われる。

20 【0112】次に、 $R < 1$ (縮小) の場合を説明する。

【0113】当該 $R < 1$ の場合、出力レジスタ25に設定するスキップ情報としては、例えば、左から i 番目の出力レジスタに対して、 $(i \bmod N_o) < N_i$ であれば通常動作し、そうでなければスキップ動作するというようなスキップ情報を設定する。ただし、 $(i \bmod N_o)$ は i を N_o で割った余りを表す。例えば、 $2/3$ 倍の縮小を行うときは、出力レジスタ25に対し、次のようなスキップ情報32が送られる(ただし○は通常動作を、×はスキップ動作を表す)。

30 【0114】○○×○○×○○×・・・

ここに、プロセッサエレメント群22から

a b c d e f g h i・・・

なるデータが入力された場合、実際には出力レジスタ25からは

a b d e g h・・・

というデータ列が出力され、データの個数は $2/3$ に減少する。

【0115】次に、上記図6のフィルタ設計回路33の具体的動作の説明を行う。

40 【0116】フィルタ設計回路33は、補間演算のための係数セットを求める。この係数セットは、水平フィルタ係数35、垂直フィルタ係数36として利用されることになる。上記補間関数としては、例えばいわゆるキュービック関数を用いることができ、このキュービック関数を用いた例について以下に説明する。

【0117】キュービック関数 $Cub(x)$ は、次式

(5)にて表されるものである。なお、式(5)は前記式(4)と同じ式である。ただし、式(5)に示されるキュービック関数の横軸は原画像をデジタル信号にサンプリングするサンプリング間隔で正規化されているもの

とする。

$$\begin{aligned} \text{Cub}(x) &= |x|^{3-2}|x|^{2+1} & (|x| \leq 1 \text{の時}) \\ \text{Cub}(x) &= -|x|^{3+5}|x|^{2-8}|x|^{+4} & (1 < |x| \leq 2 \text{の時}) \\ \text{Cub}(x) &= 0 & (2 < |x| \text{の時}) \end{aligned}$$

... (5)

この場合、フィルタのタップ数は4となり、式(6)に示すように、補間画素Yは左右2つずつの原画素(例えばD₀~D₃の4つの原画素とする)と、これら4つの原

※画素D₀~D₃に対応するフィルタ係数セットC₀~C₃の積和演算により求める。

$$Y = C_0 * D_0 + C_1 * D_1 + C_2 * D_2 + C_3 * D_3 \quad \dots (6)$$

また、上記フィルタ係数セットは、次のようにして決定する。

10★D₂と1/2だけずれている。補間画素Y₂以降については以上の繰り返しであるから、異なる2種類の係数セットが必要になる。例えば、補間画素Y₀に対応する係数セットを例えばC(0,0), ..., C(0,3)とし、補間画素Y₁に対応する係数セットをC(1,0), ..., C(1,3)とすると、

【0120】例えば、変換比率が3:2の場合、原画素Dと補間画素Yの位置関係(位相関係)は図7のようになる。

【0121】この図7において、補間画素Y₀と原画素D₀は位相が一致しており、補間画素Y₁は原画素D₁, ★

$$\begin{aligned} C(0,0) &= \text{Cub}(Y_0 \text{から見た} D_{-1} \text{の位置}) = \text{Cub}(-1) = 0 \\ C(0,1) &= \text{Cub}(Y_0 \text{から見た} D_0 \text{の位置}) = \text{Cub}(0) = 0 \\ C(0,2) &= \text{Cub}(Y_0 \text{から見た} D_1 \text{の位置}) = \text{Cub}(1) = 0 \\ C(0,3) &= \text{Cub}(Y_0 \text{から見た} D_2 \text{の位置}) = \text{Cub}(2) = 0 \\ C(1,0) &= \text{Cub}(Y_1 \text{から見た} D_0 \text{の位置}) = \text{Cub}(-3/2) = -0.125 \\ C(1,1) &= \text{Cub}(Y_1 \text{から見た} D_1 \text{の位置}) = \text{Cub}(-1/2) = 0.625 \\ C(1,2) &= \text{Cub}(Y_1 \text{から見た} D_2 \text{の位置}) = \text{Cub}(1/2) = 0.625 \\ C(1,3) &= \text{Cub}(Y_1 \text{から見た} D_3 \text{の位置}) = \text{Cub}(3/2) = -0.125 \end{aligned}$$

フィルタ設計回路33は、以上の係数セットを求めるためのプログラムを有するものである。

☆て求めた係数セットを用いて、補間画素Yを次のように計算する。

【0122】前記水平フィルタプログラム34、垂直フィルタプログラム36は、当該フィルタ設計回路33に☆

【0123】

$$\begin{aligned} Y_0 &= C(0,0) \times D_{-1} + C(0,1) \times D_0 + C(0,2) \times D_1 + C(0,3) \times D_2 \\ Y_1 &= C(1,0) \times D_0 + C(1,1) \times D_1 + C(1,2) \times D_2 + C(1,3) \times D_3 \\ Y_2 &= C(0,0) \times D_1 + C(0,1) \times D_2 + C(0,2) \times D_3 + C(0,3) \times D_4 \\ Y_3 &= C(1,0) \times D_2 + C(1,1) \times D_3 + C(1,2) \times D_4 + C(1,3) \times D_5 \end{aligned}$$

同様に、変換比率が4:3の場合の例を示す。この例では、原画素Dと補間画素Yの位相関係は3種類であるので、3種類の係数セットC(0,0), ..., C(0,3)、C(1,0), ..., C(1,3)、C(2,0), ..., C(2,3)が求められる。

◆類の係数セットが必要になる。

【0125】この4:3の場合、原画素Dと補間画素Yの位置関係(位相関係)は、図8のようになる。

【0126】ここで、当該4:3の例においても係数セットを、前述同様にC(0,0), ..., C(0,3)~C(2,0), ..., C(2,3)とすると、

【0124】この例の場合も、入出力の画素数の比をN_i:N_o(N_i, N_oは互いに素)とすると、N_o種◆

$$\begin{aligned} C(0,0) &= \text{Cub}(Y_0 \text{から見た} D_{-1} \text{の位置}) = \text{Cub}(-1) = 0 \\ C(0,1) &= \text{Cub}(Y_0 \text{から見た} D_0 \text{の位置}) = \text{Cub}(0) = 0 \\ C(0,2) &= \text{Cub}(Y_0 \text{から見た} D_1 \text{の位置}) = \text{Cub}(1) = 0 \\ C(0,3) &= \text{Cub}(Y_0 \text{から見た} D_2 \text{の位置}) = \text{Cub}(2) = 0 \\ C(1,0) &= \text{Cub}(Y_1 \text{から見た} D_0 \text{の位置}) = \text{Cub}(-4/3) = -0.148 \\ C(1,1) &= \text{Cub}(Y_1 \text{から見た} D_1 \text{の位置}) = \text{Cub}(-1/3) = 0.815 \\ C(1,2) &= \text{Cub}(Y_1 \text{から見た} D_2 \text{の位置}) = \text{Cub}(2/3) = 0.407 \\ C(1,3) &= \text{Cub}(Y_1 \text{から見た} D_3 \text{の位置}) = \text{Cub}(5/3) = -0.074 \\ C(2,0) &= \text{Cub}(Y_2 \text{から見た} D_1 \text{の位置}) = \text{Cub}(-5/3) = -0.074 \\ C(2,1) &= \text{Cub}(Y_2 \text{から見た} D_2 \text{の位置}) = \text{Cub}(-2/3) = 0.407 \\ C(2,2) &= \text{Cub}(Y_2 \text{から見た} D_3 \text{の位置}) = \text{Cub}(1/3) = 0.815 \end{aligned}$$

$$C(2,3) = \text{Cub}(Y_2 \text{から見た} D_4 \text{の位置}) = \text{Cub}(4/2) = -0.148$$

この例の場合のタイミング設計回路38の動作は以下のようになる。

【0127】当該タイミング設計回路38は垂直方向の画素数(ライン数)変換のためのタイミング情報39を生成するものである。タイミング情報39は、垂直フィルタプログラム36において、フィルタ仕様30の変換比率に基づき、入出力走査線のライン数を変換するように作用する。入出力のライン数の比を $N_i : N_o$ (N_i , N_o は互いに素)とすると、変換比率 $R = N_o / N_i$ によって拡大と縮小に動作が分けられる。

【0128】まず、 $R > 1$ (拡大) の場合を説明する。すなわち入力よりも出力の方がライン数が多い場合である。

【0129】タイミング設計回路38は、例えば i 番目の入力ラインにおいて $(i \bmod N_i) < N_o$ であればライン入力→補間演算処理→ライン出力という順序のタイミング情報39を生成し、 $(i \bmod N_i) \geq N_o$ であればライン入力を待たずにライン出力を行うので、補間演算処理→ライン出力という順序でタイミング情報39を生成する。

【0130】例えば、 $3/2$ 倍の拡大を行うとき、次のようなタイミング信号39が生成される。垂直フィルタプログラム39は、この順序に従って処理を行う。

【0131】

ライン入力→補間演算処理→ライン出力→
 ライン入力→補間演算処理→ライン出力→
 補間演算処理→ライン出力→
 ライン入力→補間演算処理→ライン出力→
 ライン入力→補間演算処理→ライン出力→
 補間演算処理→ライン出力→
 .
 .
 .

以上より、入力が2ラインにつき出力が3ラインとなり、垂直方向に $3/2$ 倍にライン数が増加する。

【0132】次に、 $R < 1$ (縮小の場合) の説明を行う。すなわち入力よりも出力の方がライン数が少ない場合である。

【0133】 i 番目の入力ラインにおいて、 $(i \bmod N_i) < N_o$ であればライン入力→補間演算処理→ライン出力という順序のタイミング情報39を生成し、 $(i \bmod N_i) \geq N_o$ であればライン入力を1回多く行うために、ライン入力のためのタイミング情報39を生成する。

【0134】例えば、 $2/3$ 倍の拡大を行うとき、次のようなタイミング信号39が生成される。垂直フィルタ*

$$Y = L_1(D) * C_0 + D * C_1 + R_1(D) * C_2 + R_2(D) * C_3 \quad \dots (7)$$

このステップST5の次のステップST6では、データ出力を行う。すなわち各要素プロセッサは、結果を下側

* プログラム39は、この順序に従って処理を行う。

【0135】

ライン入力→補間演算処理→ライン出力→
 ライン入力→補間演算処理→ライン出力→
 ライン入力→
 ライン入力→補間演算処理→ライン出力→
 ライン入力→補間演算処理→ライン出力→
 ライン入力→

.
 .
 .

以上より、入力が3ラインにつき出力が2ラインとなり、垂直方向に $2/3$ 倍にライン数が増加する。

【0136】次に、水平フィルタプログラム34の動作説明を行う。すなわち上側のプロセッサエレメント群21では、以下の図9に示すフローチャートのように水平フィルタプログラム34を実行する。

【0137】この図9において、ステップST1ではスキップ設定を行う。すなわち、スキップ情報32に従い、入力レジスタ24及び出力レジスタ25にスキップ動作の設定を行う。

【0138】ステップST2では、領域の確保を行う。すなわち、各要素プロセッサは、ローカルメモリに4タップ分の係数セット $C_0 \sim C_3$ を格納するための領域 $c_1 \sim c_4$ を確保する。

【0139】ステップST3では、係数設定を行う。例えば、 $3:2$ 縮小画素数変換の場合には、以下のようになる。例えば、左から1, 3, 5, ..., $(i+2)$ 番目の要素プロセッサは、係数セット $C(0,0)$, ..., $C(0,3)$ を係数メモリからローカルメモリに格納する。また、左から2, 4, 6, ..., $(i+1)$ 番目の要素プロセッサは、係数セット $C(1,0)$, ..., $C(1,3)$ を係数メモリからローカルメモリに格納する。

【0140】ステップST4では、データ入力を行う。各要素プロセッサは、入力レジスタ24から入力されたデータ(原画素Dのデータ)をローカルメモリの領域 d に格納する。

【0141】ステップST5では、補間演算を行う。すなわち、各要素プロセッサは、原画素Dと係数Cとを用いて、次の式(7)の演算を行い、結果を補間画素Yとして領域 y に格納する。ただし、式(7)中の $L_1(x)$, $L_2(x)$, $R_1(x)$, $R_2(x)$ (x は変数)は、それぞれ1つ左、2つ左、1つ右、1つ右の要素プロセッサのローカルメモリの領域 d の値(原画素Dの値)を表す。

【0142】

のプロセッサエレメント群22に転送する。

【0143】その後はステップST4に戻る。

【0144】次に、垂直フィルタプログラム36の動作を説明する。すなわち、下側のプロセッサエレメント群22では、以下の図10のフローチャートのように垂直フィルタプログラム36を実行する。

【0145】この図10において、ステップST11では領域を確保する。すなわち、各要素プロセッサは、ローカルメモリに4ライン分の原画素D用の領域d1~d4を確保する。

【0146】ステップST12では各要素プロセッサが、タイミング情報39に従って以下の(A)から(C)の動作を選んで実行する。

【0147】このステップST12にて(A)が選ばれると、ステップST13にてライン入力が行われ、上側のプロセッサエレメント群21からの入力データ(原画素D)をローカルメモリの領域(d0~d3とする)に格納*

$$Y = D_0 * C_0 + D_1 * C_1 + D_2 * C_2 + D_3 * C_3 \quad \dots (8)$$

さらに、ステップST12にて(C)が選ばれたとき、ステップST19にてライン出力を行う。すなわち、各要素プロセッサは、領域yの値(補間画素Y)を出力レジスタ25に格納する。

【0151】上述したようなことから、本発明実施例の画像信号処理装置によれば、画素数変換の処理を全て自動的に行うことができ、フィルタの仕様を決定するだけで、その仕様に基づく画素数変換装置が実現できる。また本実施例の装置によれば、プログラムや係数を入れ換えるだけで、回路の変更をすることなく、多種の画素数変換を含むフィルタ処理が行え、さらに、分離型フィルタを利用することで効率的な水平、垂直フィルタを一般の1次元フィルタ設計手法により実現でき、また、フィルタ処理を2つの独立したプロセッサエレメント群で実行するという構成により、プログラムの独立性、保守性などを高めることができ、利用し易くなっている。

【0152】

【発明の効果】本発明においては、フィルタ仕様情報に基づいて第1、第2の両方向用のフィルタ演算に必要なフィルタプログラム及びフィルタ係数を発生し、第1の方向の画素順で順次シリアルに入力した画素データをパラレルに変換し、このパラレル変換された第1の方向の画素列の各画素データに対して第1の方向用のフィルタプログラム及びフィルタ係数を用いた所定のフィルタ演算処理を施し、その後の第1の方向の画素列の各画素データに対して、必要に応じて更に第2の方向用のフィルタプログラム及びフィルタ係数を用いた所定のフィルタ演算処理を施し、得られたパラレル画素データを第2の方向の画素順に順次シリアル出力することにより、例えばタップ数や変換比率を変更したい時に、容易かつ短時間でそれらの変更が可能であり、また、複数の変換比率にて画素数変換を行う必要がある場合でもそれぞれの変換比率に対応する複数用意する必要がなくなる。

【0153】すなわち、本発明の画像信号処理方法及び

*納する。最新の4ライン分のデータを保持するため、以下の手順を実行する。

【0148】ステップST14では領域d0に領域d1の原画素Dの値をコピーする。ステップST15では、領域d1に領域d2の原画素Dの値をコピーする。ステップST16では領域d2に領域d3の原画素Dの値をコピーする。ステップST17では領域d3に入力データ(原画素D)の値をコピーする。

【0149】また、ステップST12にて(B)が選ばれると、ステップST18にて補間演算を行う。すなわち、原画素Dと係数Cとを用いて次の式(8)の演算を行い、結果を補間画素Yの値として領域yに格納する。なお、フィルタ係数は全要素プロセッサで共通なので、係数メモリから直接読み出す。

【0150】

装置においては、フィルタプログラム及びフィルタ係数をフィルタ仕様情報に基づいて全て自動的に設定可能であり、したがって、回路の変更をすることなく、フィルタの仕様を決定するだけで、プログラムやフィルタ係数を入れ換えることができ、多種の画素数変換や走査線変換を含むフィルタ処理が実現可能である。また、本発明においては、2次元画像を第1の方向と第2の方向に分けて考え、これら第1の方向と第2の方向のそれぞれに対して分離してフィルタ演算を行うようにしているため、フィルタ演算のためのフィルタプログラム及びフィルタ係数を容易かつ効率的に設計可能である。さらに、本発明においては、このようにフィルタ演算を分離することで、第1の方向と第2の方向のフィルタ演算の独立性や保守性を高めることができ、利用しやすくなる。

【図面の簡単な説明】

【図1】本発明実施例のリニアアレイ型多並列プロセッサの基本構成を示すブロック回路図である。

【図2】リニアアレイ型多並列プロセッサの要素プロセッサの単一エレメントの基本構成を示すブロック回路図である。

【図3】リニアアレイ型多並列プロセッサの要素プロセッサの具体的構造を示す回路図である。

【図4】本発明実施例のリニアアレイ型多並列プロセッサをより詳細に示すブロック回路図である。

【図5】画素数変換時の補間画像演算の説明に用いる図である。

【図6】本発明実施例のリニアアレイ型多並列プロセッサにおけるプログラムとフィルタ係数の自動設計の流れを示す図である。

【図7】3:2画素数変換時の原画素と補間画素の位置関係の説明に用いる図である。

【図8】4:3画素数変換時の原画素と補間画素の位置関係の説明に用いる図である。

【図9】水平フィルタプログラムを実行する際の流れを

31

示すフローチャートである。

【図10】垂直フィルタプログラムを実行する際の流れを示すフローチャートである。

【図11】画像の拡大と標準化周波数（画素数）の変換についての説明に用いる図である。

【図12】画像の縮小と標準化周波数（画素数）の変換についての説明に用いる図である。

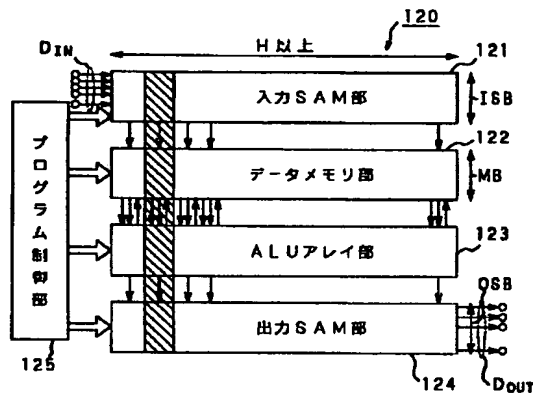
【図13】原画像には存在しなかった位置の画素データ値を求める補間フィルタ演算についての説明に用い、水平方向の画素の位置関係を模式的に示す図である。

【図14】理想的な補間関数と近似された補間関数の説明に用いる図である。

【図15】従来のハードウェア構成による画像信号処理装置の一例を示すブロック回路図である。

【図16】従来のハードウェア構成による画像信号処理装置の各部の動作説明に用いる表である。

【図1】



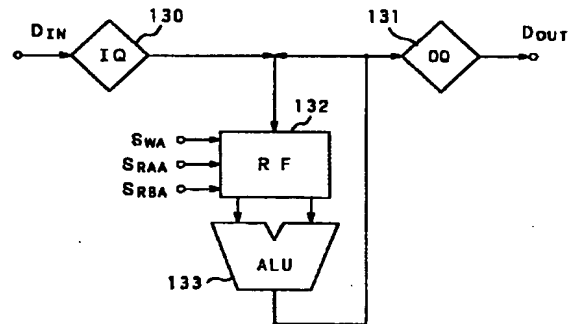
32

【図17】従来のハードウェア構成による画像信号処理装置にて扱う係数等の説明に用いる表である。

【符号の説明】

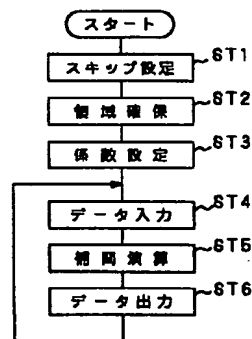
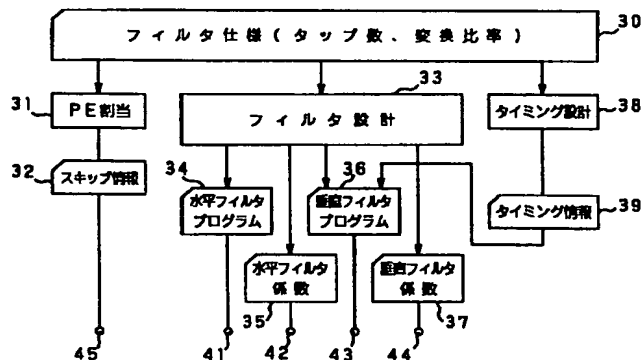
20, 120 リニアアレイ型多並列プロセッサ、 1
21 入力SAM部、 122 データメモリ部、 123
ALUアレイ部、 124 出力SAM部、 125
プログラム制御部、 130 入力バッファメモリ、
131 出力バッファメモリ、 132 データメモ
リ、 133, 123B ALU、 121A 入力ボイ
ンタ、 121B 入力SAMセル、 122A メモ
リセル、 123A ALUセル、 124A 出力ボ
インタ、 124B 出力SAMセル、 21, 22
プロセッサエレメント群、 23 要素プロセッサ、 2
4 入力レジスタ、 25 出力レジスタ、 26, 2
8 コントローラ、 27, 29 係数メモリ

【図2】

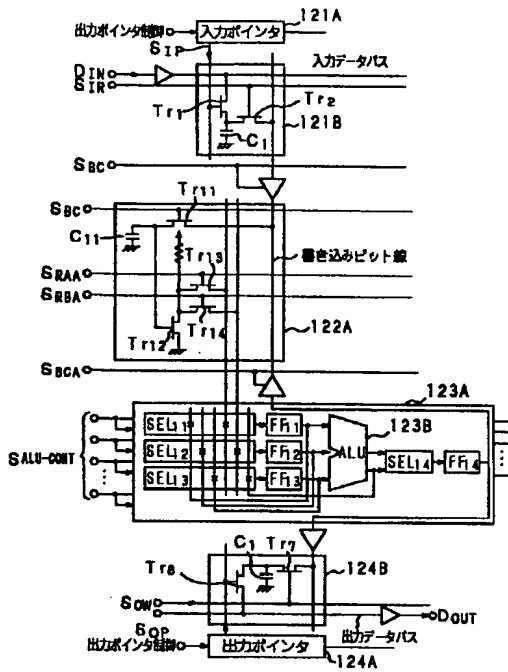


【図9】

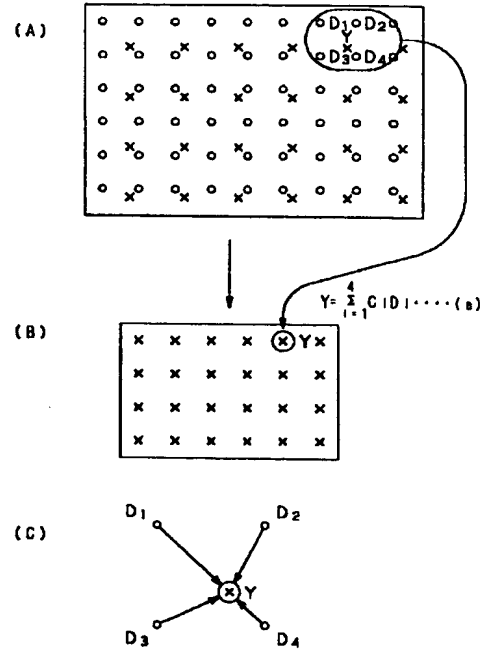
【図6】



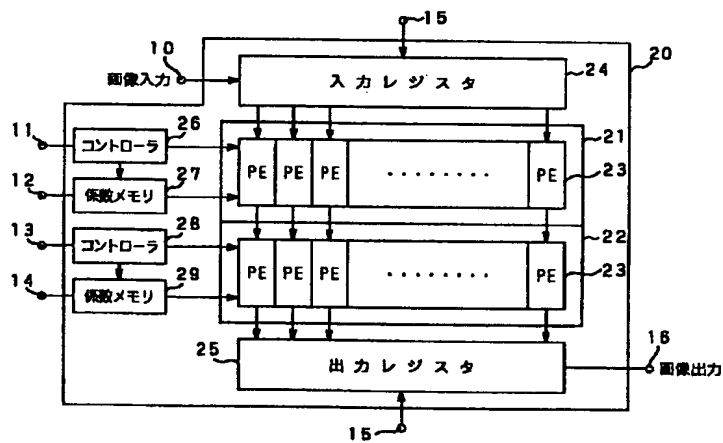
【図3】



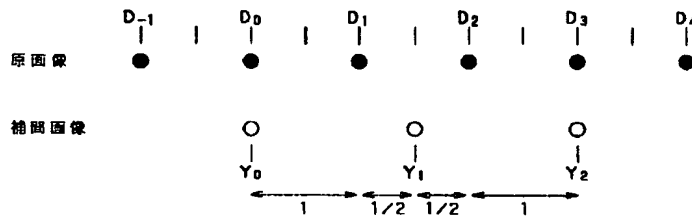
【図5】



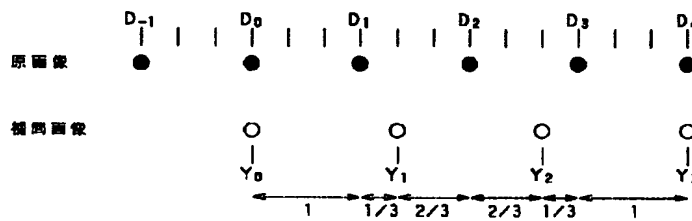
【図4】



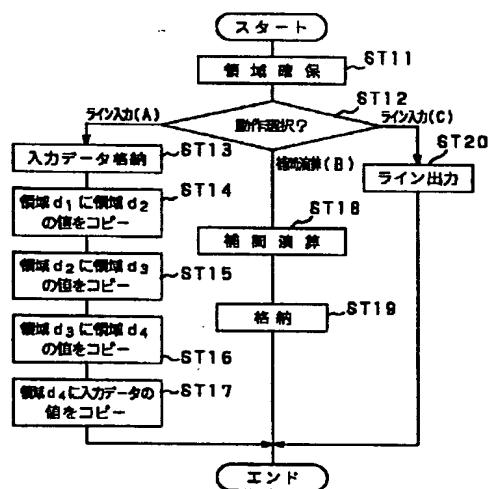
【図7】



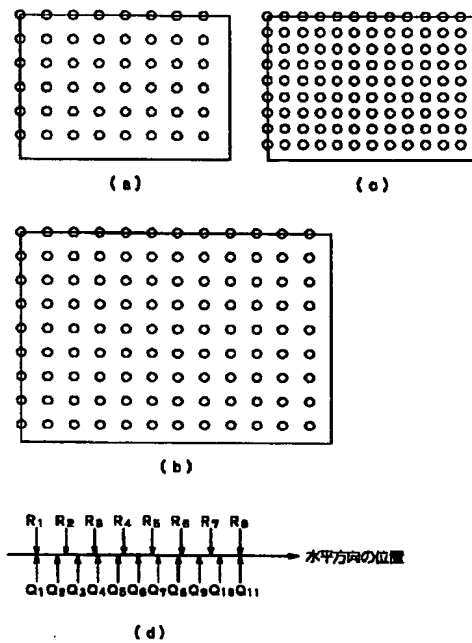
【図8】



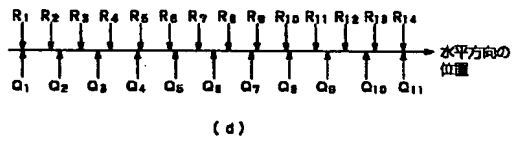
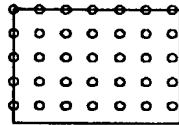
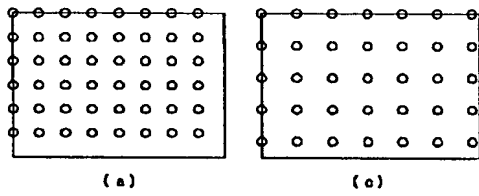
【図10】



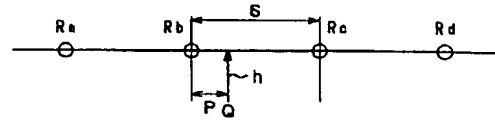
【図11】



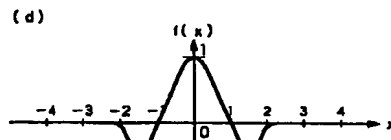
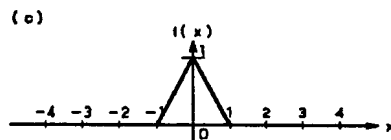
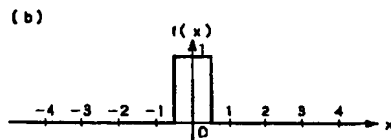
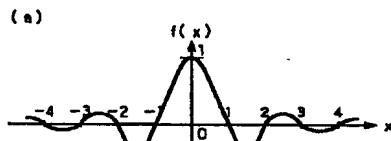
【図12】



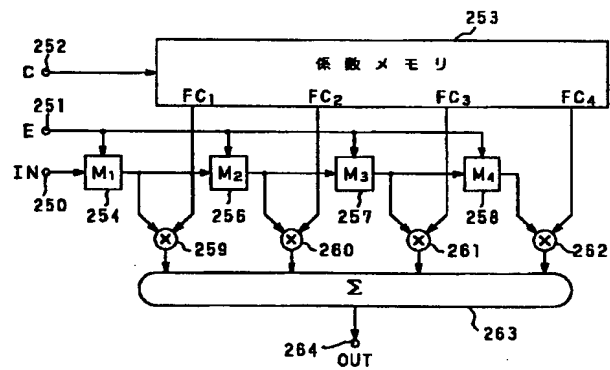
【図13】



【図14】



【図15】



【図16】

サイクル	IN	C	E	M ₁	M ₂	M ₃	M ₄	OUT
1	R ₁		H	R _{m0}	R _{m1}	R _{m2}	R _{m3}	
2	R ₂		H	R ₁	R _{m0}	R _{m1}	R _{m2}	
3	R ₃		H	R ₂	R ₁	R _{m0}	R _{m1}	
4	R ₄	P ₀	L	R ₃	R ₂	R ₁	R _{m0}	Q ₁
5	R ₄	P ₇	H	R ₃	R ₂	R ₁	R _{m0}	Q ₂
6	R ₅	P ₄	H	R ₄	R ₃	R ₂	R ₁	Q ₃
7	R ₅	P ₁	L	R ₅	R ₄	R ₃	R ₂	Q ₄
8	R ₅	P ₈	H	R ₅	R ₄	R ₃	R ₂	Q ₅
9	R ₇	P ₅	H	R ₆	R ₅	R ₄	R ₃	Q ₆
10	R ₈	P ₂	L	R ₇	R ₆	R ₅	R ₄	Q ₇
11	R ₈	P ₉	H	R ₇	R ₆	R ₅	R ₄	Q ₈
12	R ₉	P ₆	H	R ₈	R ₇	R ₆	R ₅	Q ₉
13	R ₁₀	P ₃	L	R ₉	R ₈	R ₇	R ₆	Q ₁₀
14	R ₁₀	P ₀	H	R ₉	R ₈	R ₇	R ₆	Q ₁₁

【図17】

位相 P _i	浮動小数点表現係数				8ビット表現係数				総数値
	FC ₁	FC ₂	FC ₃	FC ₄	FC ₁	FC ₂	FC ₃	FC ₄	
P ₀ = $\frac{0}{10}$ = 0.0	0.0	1.0	0.0	0.0	0	128	0	0	128
P ₁ = $\frac{1}{10}$ = 0.1	-0.009	0.981	0.109	-0.081	-1	126	14	-10	128
P ₂ = $\frac{2}{10}$ = 0.2	-0.032	0.928	0.232	-0.128	-4	119	30	-16	129
P ₃ = $\frac{3}{10}$ = 0.3	-0.063	0.847	0.363	-0.147	-8	108	46	-19	127
P ₄ = $\frac{4}{10}$ = 0.4	-0.086	0.744	0.496	-0.144	-12	95	63	-18	128
P ₅ = $\frac{5}{10}$ = 0.5	-0.125	0.625	0.625	-0.125	-16	80	80	-16	128
P ₆ = $\frac{6}{10}$ = 0.6	-0.144	0.496	0.744	-0.096	-18	63	95	-12	128
P ₇ = $\frac{7}{10}$ = 0.7	-0.147	0.363	0.847	-0.063	-19	46	108	-8	127
P ₈ = $\frac{8}{10}$ = 0.8	-0.128	0.232	0.928	-0.032	-16	30	119	-4	129
P ₉ = $\frac{9}{10}$ = 0.9	-0.081	0.109	0.981	-0.009	-10	14	126	-1	128

フロントページの続き

(72)発明者 青山 幸治
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

THIS PAGE BLANK (USPTO)